

DORSEY & WHITNEY LLP
Sri Sankaran (SBN 236584)
555 California Street, Suite 1000
San Francisco, CA 94104-1513
Telephone: (415) 781-1989
Facsimile: (415) 398-3249

Devan V. Padmanabhan (Admitted *pro hac vice*)
Suite 1500, 50 South Sixth Street
Minneapolis, MN 55402-1498
Telephone: (612) 340-2600

Attorneys for Defendant
Sproqit Technologies, Inc.

E-FILING

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

Visto Corporation

Plaintiff,

v.

Sproqit Technologies, Inc.

Defendant.

) CASE NO.: 04-0651 EMC
)
) Before the Honorable Edward M. Chen
)

) **DECLARATION OF STEPHEN**
) **WEINSTEIN IN OPPOSITION TO**
) **PLAINTIFF'S MOTION FOR**
) **PRELIMINARY INJUNCTION**
)

) **Hearing Date:** January 18, 2005
) **Time:** 10:30 a.m.
) **Courtroom:** Courtroom C, 15th Floor
) **Judge:** Edward M. Chen
)

I, Stephen Weinstein, hereby declare:

INTRODUCTION

1. I am a consultant with my own personal consulting company, Communication Theory and Technology Consulting LLC, and a long-time member of the communications engineering community. I have a Ph.D. in Electrical Engineering from the University of

1 California-Berkeley. I have 40 years of experience largely in industrial research and research
2 management at Bell Laboratories, Bellcore, and the NEC-Princeton laboratory, but also
3 including five years as VP-Technology Strategy at the American Express Company. I was
4 President of the IEEE Communications Society in 1996-97 and a member of the IEEE Board of
5 Directors in 2002-2003. I am the 2006-2007 Chief Technology Officer of the IEEE
6 Communications Society responsible for development of the Society's electronic services. I am
7 an Adjunct Professor of Electrical Engineering at Columbia University where I taught a graduate
8 course in Integrated Networking in Fall 2005. I am author or coauthor of three books: Getting
9 The Picture, A Guide to the New Electronic Media (IEEE Press 1984); Data Communication
10 Principles (coauthored), Plenum 1992; and The Multimedia Internet (a light technical
11 explanation of communications networking and technologies, media compression, and Internet
12 protocols and services supporting audio/video media on the Internet), Springer 2005.

13 2. I have been retained by Dorsey & Whitney LLP to render certain technical
14 opinions with regard to the infringement and validity of claim 10 of U.S. Patent No. 6,085,192
15 ("the '192 Patent"). First, I have been asked to consider whether Sproqit's Personal Edition and
16 Work Group Edition ("the Sproqit system") infringes claim 10. Second, I have been asked to
17 consider whether claim 10 is valid in view of certain systems and publications that pre-date the
18 patent application that issued as the '192 Patent ("the '192 Patent Application") by more than
19 one year.

20 3. I have reviewed the '192 Patent, including the written description and the claims;
21 the file history of the '192 Patent and the cited references; as well as claim 10 as issued during
22 re-examination and certain parts of the re-examination file history. I have also reviewed selected
23 pleadings in this case, including the Declaration of Dr. Sabin Head, Visto's motion for
24 preliminary injunction, and Exhibit B to the Khaliq Declaration, which I understand is the claim
25 construction ruling from the Texas litigation. I have also reviewed the Declaration of Barry

1 Roitblat submitted by Sproqit in opposition to Visto's motion for preliminary injunction, and I
 2 have met with Mr. Roitblat to observe the operation of the Sproqit system and to discuss the
 3 technical aspects of the Sproqit system.

4 4. A copy of my CV is attached as Exhibit A.

5 **SUMMARY OF OPINIONS**

6 5. It is my opinion that the Sproqit system (by which I mean both Sproqit Personal
 7 Edition and Sproqit Work Group Edition) does not infringe claim 10 of the '192 Patent for
 8 reasons that include at least the following:

- 9 • the Sproqit system does not make a "copy" of a document (e.g., an e-mail) for
 10 use at a handheld unit – rather it only makes portions of the document (e.g., an e-
 11 mail) available at the handheld unit;
- 12 • the Sproqit system does not compare versions of a document to select a preferred
 13 version – rather the Sproqit system uses a very old technique in which changes
 14 are simply pushed out to the server unit or the handheld unit;
- 15 • because of this fundamental difference in approach – pushing a queue of actions
 16 rather than an approach where a comparison is made between two documents –
 17 the Sproqit system does not have the following structures required by the claim:
 18 "a general synchronization module," "a synchronization agent," and "a
 19 synchronization-start module".

20 6. It is my opinion that claim 10 is invalid for at least the following reasons:

- 21 • the Coda system described in publications and implemented at Carnegie Mellon
 22 University in the 1990-1991 time period renders claim 10 obvious; and
- 23 • Lotus Notes as described in publications and made available to the public in the
 24 early 1990s renders claim 10 obvious.

25 Neither of these appear to have been considered by the Patent Office.

INTERPRETING PATENTS

8. **Claim Construction**: I understand that terms in a patent claim typically are given the meaning that they would have to one of ordinary skill in the art at the time the invention was made. In this regard, I understand, one must review the patent, including the claims and written description, as well as the patent's file history, as these may influence the meaning that particular claim terms might have to one of ordinary skill in the art. I understand that where a claim term has an ordinary meaning to a person of ordinary skill in the art, that is the meaning that is generally given to the term, unless the claims, written description, or file history require that the term be given a different meaning.

4

1 claim is not found in the accused device, either literally or under the doctrine of equivalents, then
2 the claim is not infringed.

3 10. **Invalidity**: I also understand that if one reference shows all elements of a claim,
4 then such a claim is anticipated by the reference. It is my further understanding that the prior art
5 can specifically disclose a claim element or that the disclosure can be inherent in the prior art. I
6 understand that a patent claim is obvious, even if it is not identically disclosed in a single prior
7 art reference, if the differences between the subject matter claimed and the prior art are such that
8 the subject matter claimed would have been obvious to one of ordinary skill in the art at the time
9 the claimed invention was made.

10 **THE SPROQIT SYSTEM**

11 11. The Sproqit system is described in detail in the Declaration of Barry Roitblat. I
12 have relied on Mr. Roitblat's description of the Sproqit system. As a general summary, the
13 Sproqit technology uses a traditional approach to managing documents between a central unit
14 and one or more remote units. This traditional approach is a different technology from that
15 described in '192 Patent. This architecture — of a central unit, which is called a server, and
16 remote units, each called a "client,"— is known as the client-server architecture. In this
17 architecture, clients may be connected (via a physical or wireless link) to the server. In some
18 applications, there are periods when the client is connected to the server, and there are periods
19 when the clients are disconnected from the server. In managing a traditional client-server
20 architecture, during periods of disconnection, a series of actions on either side is queued up and
21 then transferred and executed when connection is established. This approach of queuing up
22 actions for transmission without request when connected is an example of "push" technology.
23 This is one of the oldest architectures used in computing, and a different approach from that
24 taken in Claim 10.

1 12. This traditional approach employed in the Sproqit system is a different technology
2 from that described in the '192 Patent. That is, the '192 Patent requires:

- 3 • a "copy" of the document to exist at the server and at the remote unit;
- 4 • a comparison of version information to select a preferred version; and
- 5 • that the server initiate such a comparison.

6 In contrast, in the Sproqit system,

- 7 • there is no copy of a file,
- 8 • there is no comparison of version information,
- 9 • there is no selection of a preferred version; and
- 10 • there is no initiation of a comparison from the server

11 Sproqit's "value added" is in user interface and application interface features outside the scope of
12 the '192 Patent.

13 13. As an example, in the prior art of this traditional client-server technique for
14 "pushing" queued actions from one entity to another, consider its early use (along with additional
15 synchronization techniques) in one aspect of the Coda system for intermittently-connected
16 computing implemented in the 1980s and early 1990s at Carnegie Mellon University, which is
17 discussed in more detail below. The Sproqit system is limited to the simple queued actions
18 model.

19 14. In summary, in the Sproqit system, the hand held operates as a remote control for
20 the server PC. In the Sproqit system, a workspace element, such as an email, exists on the server
21 only. An email consists of a large number of fields. There is no copy of the email on the hand
22 held device since only a small selected subset of fields, not a copy of the complete email
23 message, is pushed from the server to the hand held. Many fields in email messages are never
24 sent to the hand held device. Among the e-mail fields that are never forwarded to the hand held
25 device in the Sproqit system is the field for importance, the field for Unique ID, the field for

categories and the field for reply to lists. The Sproqit system pushes (meaning sends without waiting for a request) queued events independently in both directions (client to server, server to client) that represent changes to fields that are a part of an email message. These events are processed as received, as in classical client-server systems. The Sproqit system does not compare values or versions. There is no synchronization engine, just transmission of queued events.

NONINFRINGEMENT ANALYSIS

15. I have carefully reviewed both the content of '192 Patent and the technology implemented in Sproqit system and I am convinced that the Sproqit technology is entirely different from that described in the claims of the '192 Patent for at least the reasons set forth below. I believe, and will describe below, exactly where the declaration of Dr. Head, made on behalf of the plaintiff Visto, has overlooked some critical differences between Claim 10 and the Sproqit technology.

16. Dr. Head attempts to demonstrate infringement by using an e-mail as a “workspace element” of Claim 10. In addressing Dr. Head’s analysis, I will assume that an e-mail is a workspace element.

In the Sproqit System There Is No “Copy” Of a Workspace Element On a Smart Phone

17. Claim 10 requires “an independently modifiable copy of the first workspace element at a second store on a smart phone. . . .” (emphasis added). It is my opinion that the Sproqit system does not meet this claim limitation literally or under the doctrine of equivalents.

Construction of “workspace element”

18. **Intrinsic Evidence:** The patent teaches that a workspace element is a subset of workspace data, such as an email.

It will be appreciated that the e-mail data 165, file data 170,

1 calendar data 175 and user data 180 or portions thereof may be
2 stored at different locations such as locally on the desktop
3 computer. It will be further appreciated that the email data 165,
4 file data, 170, calendar data 155 and user data 180 are exemplary
5 and collectively referred to herein as “workspace data” 185. Those
6 skilled in the art will recognize that “workspace data” may include
7 other types of data such as application programs. It will be further
8 appreciated that the e-mail data 165, file data 170, calendar data
9 175 and user data 180 may each be divided into workspace
10 elements, wherein each workspace element is identified by
11 particular version information 255 (described below with reference
12 to Fig. 2). **Accordingly, each e-mail, file, calendar, etc. may be**
13 **referred to as “a workspace element in workspace data.”**

14 ‘ 192 Patent, Column 3, Lines 16-31 (emphasis added).

15 19. **Texas Litigation:** In the Texas Litigation Judge Ward construed the term
16 “workspace element as “a subset of workspace data such as an e-mail, file, bookmark, calendar,
17 or applications program which may include version information.” Khaliq Ex. A, at 22.

18 20. **Dr. Head’s Construction:** Dr. Head relies on the construction of “workspace
19 element” in the Texas Litigation. Dr. Head further defines “e-mail”: “The term e-mail refers to
20 the message headers, the message body, status indicators (such as (opened/unopened,
21 deleted/undeleted, and other parameters such as urgency, and other message or system dependent
22 parameters, etc.).” Head Decl., Page 17, Lines 5-7. For the purpose of this motion, I will
23 assume that Dr. Head’s construction of workspace element and his definition of e-mail are
24 appropriate.

25 **Construction of “copy”:**

21. **Intrinsic Evidence:** The patent teaches that a “copy” should include all the elements of the original. For example, the Patent teaches that a “copy” of a workspace element should include all the components of the original:

An independently modifiable copy of the workspace data 185, referred to herein as workspace data 123, is stored on the global server 120 for easy access by a user from the remote terminal 105. Being a copy, the workspace data 123 includes independently modifiable copies of each workspace element in workspace data 185 and an independently modifiable copy of version information 255 (Fig. 2), referred to herein as version information 124.

‘192 Patent, Column 3, Lines 32-40. Accordingly, just as the patent teaches that a “copy” of workspace data must include all the workspace elements contained in the data, a “copy” of the workspace element must include all the data contained in the workspace element.

22. **Texas Litigation:** In the Texas litigation, Judge Ward considered the meaning of the term “independently modifiable copy.” Khaliq Decl. Ex. A at 16-17. Judge Ward rejected Visto’s proposed definition that “[t]he copy does not have to be an exact copy,” finding that “Visto’s proposed definition appears to suggest a broader meaning of copy than the claim language supports.” *Id.* at 17. Judge Ward found, the written description of a related patent, the ‘708 Patent, would support a definition that the “copy of the workspace element does not have to be in the same format as the workspace element.” *Id.* at 17.

23. **Extrinsic Evidence:** The normal understanding in the information technology industry is that a copy of a data file is identical in every data bit to the original file.

Comparison of Claim and Accused System

24. In the Sproqit system there is no copy of a workspace element on a smart phone. In the Sproqit system, the workspace element – the e-mail Dr. Head’s example – exists on the

1 server only. An email consists of a large number of fields. Only a small, selected subset of
2 those fields is pushed from the server to the hand held. There is no copy of the e-mail on the
3 hand held device since many fields in e-mail messages are never sent to the hand held device.
4 One of the e-mail fields that is never forwarded to the hand held device in the Sproqit system is
5 the field for urgency or importance, which Dr. Head specifically identified as being a part of the
6 e-mail. Head Decl., ¶ 28, page 17. Other e-mail fields that are never forwarded to the hand held
7 device in the Sproqit system are the fields for flag-for-follow-up, importance, Unique ID,
8 categories and reply to lists. Changes to any of the fields not pushed to the hand held device will
9 not be pushed to and will never be represented on or written to the hand held.

10 25. Dr. Head's screen shots appear to confirm that there is no copy of the workspace
11 element on the hand held device – the screen shots from his desktop computer show a flag, but
12 that flag does not appear in any of the screen shots from the hand held device. Exhibit A to the
13 Roitblat Declaration similarly confirms that in the Sproqit system, importance and flag-for-
14 follow-up fields that exist and are displayed on the desktop computer are not forwarded and are
15 not shown on the handheld device. When an email message is marked as “urgent” or is
16 “flagged-for-follow-up,” those indicators appear in Microsoft Outlook on the desktop computer.
17 But that data is not pushed to the hand held device and the urgency and flag-for-follow-up icons
18 do not appear on the hand held device.

19 26. To support his conclusion that this claim limitation is met in the Sproqit system,
20 Dr. Head relies on his Test 3. Head Decl., page 18, column 25 – page 19, column 1. In his Test
21 3, Dr. Head starts with the hand held device disconnected. A message is received at the inbox on
22 the desktop PC. Dr. Head then connects the hand held device and observes that an unopened e-
23 mail icon appears on both the PC and on the hand held device. Dr. Head assumes incorrectly
24 that because an e-mail icon appears on the hand held that a copy of the email exists on the hand
25

1 held device. Rather, as described above, just a few fields from the e-mail have been pushed to
2 the hand held device.

3 27. To summarize the Sproqit system does not infringe because Claim 10 requires a
4 copy of a workspace element on a second store at a smart phone, and in the Sproqit system there
5 is no such copy and nothing substantially equivalent to such a copy.

6 **The Sproqit system does not “compare” first and second version information.**

7 28. Claim 10 requires “means for generating a preferred version from the first
8 workspace element and from the copy by comparing the first version information and the second
9 version information.” It is my opinion that the Sproqit system does not meet this claim
10 limitation literally or under the doctrine of equivalents.

11 **Construction of “comparing”**

12 29. **Intrinsic Evidence:** The patent teaches that “comparing” first and second
13 version information means, at a minimum, that the first and second version information are
14 brought together so that their similarities or differences are evaluated. In the system of Claim 10,
15 the comparison is performed as follows:

16 the synchronization start module initiates the general
17 synchronization module and the synchronization agent;
18 the general synchronization module examines first version
19 information to determine if the first workspace element has been
20 modified;
21 the synchronization agent forwards second version information to
22 the general synchronization module;
23 finally the system has “means” which compares the first and
24 second version information to generate the preferred version.

1 30. The '192 Patent further teaches that synchronization is accomplished by
2 comparing first and second version information. A number of software modules are described
3 and claimed in Claim 10 for performing this comparison:

4 The present system includes a general synchronization module at
5 the client site for operating with a first firewall and for examining
6 first version information to determine whether a first workspace
7 element has been modified. The system further includes a
8 synchronization agent at the global server for operating outside the
9 first firewall and for forwarding to the general synchronization
10 module second version information which indicates whether an
11 independently modifiable copy of the first workspace element has
12 been modified. A synchronization-start module, at the client site,
13 operates within the first firewall and initiates the general
14 synchronization module and the synchronization agent when
15 predetermined criteria have been satisfied. The system further
16 includes means for generating a preferred version from the first
17 workspace element and from the copy by comparing the first
18 version information and the second version information, and
19 means for storing the preferred version at the first store and at the
20 second store.

21 '192 Patent, Column 1, Line 60 – Column 2, Line 10.

22 The synchronization start module 420 includes routines for
23 determining when to initiate synchronization of workspace data
24 123 and workspace data 185. * * * The synchronization-start
25 module 420 initiates data synchronization by instructing the

1 general synchronization module 425 to begin execution of its
2 routines.

3 The general synchronization module 425 includes routines for
4 requesting version information 124 from the synchronization agent
5 126 (Fig. 1) and routines for comparing the version information
6 255 against a last synchronization signature 435 such as a last
7 synchronization date and time to determine which versions have
8 been modified. **The general synchronization module 425**
9 **further includes routines for comparing the version**
10 **information 124 and the version information 255 to determine**
11 **whether if only one or both versions of a particular workspace**
12 **element have been modified and routines for performing an**
13 **appropriate synchronizing responsive action.**

14 ‘192 Patent Column 5, Lines 35-61 (emphasis added).

15 31. **Texas Litigation:** The claim construction in the Texas litigation does not shed
16 light on the meaning of the word “comparing.” However, the Texas litigation defined the entire
17 “means” for performing this “comparing” function to be the general synchronization module.

18 **Comparison of Claim and Accused System**

19 32. Claim 10 requires means for generating a preferred version from the first
20 workspace element and the copy by comparing first and second version information. As
21 described, in the Sproqit system, there is no copy of a workspace element. Moreover, in the
22 Sproqit system, there is no comparison of first and second version information to generate a
23 preferred version. Rather, events are queued and pushed from the remote device to the desktop
24 computer and vice versa. The Sproqit system does not compare version data to decide whether
25

1 an original and a copy are different or to generate a preferred version. Rather, changes or events
2 are automatically pushed as they happen, and are acted upon by the recipient.

3 33. To support his opinion that this limitation is met in the Sproqit system, Dr. Head
4 relies on his Tests 6 and 7. In Dr. Head's Test 6, he disconnects the Sproqit hand held and marks
5 an e-mail as unread on the computer. He observes that the e-mail now appears as unread on the
6 computer, and as read on the disconnected hand held. In Test 7, Dr. Head reconnects the hand
7 held and notes that the change from read to unread "has been propagated to the Sproqit
8 Companion." From these observations, Dr. Head inferred incorrectly that "software modules
9 within the Sproqit Desktop Agent compare the first and second version information and generate
10 a preferred version which replaces the version of that e-mail on the Sproqit Companion."

11 34. In fact, all that happened in Dr. Head's Tests 6 and 7 is that an event was queued
12 at the desktop during the time that the hand held was disconnected. When the hand held was
13 reconnected, the computer pushed that event to the hand held. In particular, the desktop sent a
14 message to the hand held instructing the hand held to set the icon for that message to the picture
15 of an opened envelope. (Although the desktop stores a True-False status field called "UnRead",
16 there is no such field for read/unread status on the hand held, there is only an icon which may
17 represent a variety of information including whether the message is a mail item or meeting
18 request and whether it has attachments.) There was no comparison of version information and
19 no generation of a preferred version involved. All of Dr. Head's tests are the result of the
20 operation of the Sproqit system as described. All of Dr. Head's observations are the result of
21 pushing "events" that are in the queue.

22 35. Dr. Head observed a single change from read to unread, noted that this change
23 was "propagated" from desktop to handheld, and assumed that version information was
24 compared to achieve this result. The Roitblat Declaration describes two further experiments
25 which shows that Dr. Head is incorrect. The experiments performed by Mr. Roitblat involves the

1 operation of the Sproqit system when multiple changes are made to the same message when the
2 hand held is disconnected helps show that no comparison and no synchronization or generation
3 of a preferred version is involved.

4 36. In the first experiment (shown in the video file "Demo1.exe" on Exhibit B to the
5 Roitblat Declaration) a message initially appears as unread in both the desktop and the hand held.
6 The hand held is disconnected. On the desktop computer the message is marked as read then
7 unread a number of times sequence. As each change is made, an event is queued at the desktop
8 for transmission to the hand held. The hand held is reconnected. Instead of comparing the two
9 "versions" — the last "version" from the hand held and the current "version" on the desktop —
10 and selecting a preferred version, the Sproqit system simply processes the events in sequence.
11 As each event is pushed to the hand held, the message icon on the hand held device changes. In
12 this example, the message on the hand held starts as unread (a sealed envelope icon) then
13 changes to read (an open envelope icon) when the first queued event arrives. As the subsequent
14 queued events arrive the icon cycles between read and unread icons, ending at unread when the
15 last queued event arrives. Events are pushed and acted on, but there is no comparison or
16 generation of a preferred version.

17 37. In the second experiment, (shown in the video file "Demo2.exe" on Exhibit B to
18 the Roitblat Declaration), an open e-mail appears on both the desktop computer and the
19 handheld. The hand held is disconnected, and a series of changes is made to the content of the
20 email on the desktop computer. The content of the message is changed from "1. This is the
21 original message" to "2. This is the message after a change" and saved. Then, the content of the
22 message is changed from "2. This is the message after a change" to "3. This is the message
23 after yet another (second) change" and saved. Then, the content of the message is changed from
24 "3. This is the message after yet another (second) change" and to "4. This is now the message
25 after the third and final change" and the hand held is reconnected. With each change a return

1 was added for ease of viewing. The video file shows the content of the message on the handheld
2 cycle through the values as it processes the change events as they are received.

3 38. These demos highlight the event-driven nature of the Sproqit architecture. In both
4 cases, multiple changes are made to an e-mail message while the hand held is disconnected from
5 the desktop computer. In a synchronization environment as outlined in the '192 Patent, one
6 would expect the synchronization engine to compare the current state of the messages at the time
7 the desktop computer and hand held reconnect and synchronize them by writing the "preferred
8 version" to both places. From these demos, however, one can see that there is no
9 comparison/synchronization taking place. Instead, each of the changes is causing an event to be
10 queued. Each event is then processed at the other end.

11 39. As shown in these demonstrations, the Sproqit system does not generate a
12 preferred version by comparing version information.

13 40. In summary, the Sproqit system does not infringe Claim 10 because the Sproqit
14 system does not have means for comparing first and second version information to generate a
15 preferred version, or anything substantially equivalent to such means.

16 **INVALIDITY ANALYSIS**

17 **Coda Invalidates Claim 10**

18 41. As set forth in greater detail in the claim chart attached as Exhibit B to this
19 Declaration, the Coda computing environment from the early 1990's makes Claim 10 of the '192
20 Patent invalid as obvious.

21 42. "Andrew" [which was later renamed "Coda"] is a distributed computing
22 environment that has been under development at Carnegie Mellon University since 1983." M.
23 Satyanarayanan, "Integrating Security in a Large Distributed System," ACM Transactions on
24 Computer Systems, Vol. 7, No. 3, August 1989, pp. 247-280 ("Integrating Security") (attached
25 as Exhibit E) at 247. One aspect of Coda as implemented at Carnegie Mellon is "disconnected

operation” in which a file that a client has obtained from a server is worked on independently by the client during the period of disconnection. James J. Kistler and M. Satyanarayanan, “Disconnected Operation in the Coda File System, ACM Transactions on Computer Systems”, Vol. 10, No. 1, February 1992, pp. 3-25 (“KS”) (attached as Exhibit D) at 3. Disconnected operation is a mode of operation that occurs when a client is disconnected from the server.

43. In Coda, “[a]n important . . . application of disconnected operation is in supporting portable computers.” KS (Ex. D) at 3. The KS paper shows “that disconnected operation is feasible, efficient and usable by describing its design and implementation in the Coda File System. Id. at 3.

44. Andrew/Coda has been operating at Carnegie Mellon since the 1980s. In 1990, disconnected operation was demonstrated in the Coda system. That is, in 1990, portable computers were able to connect to a server and work on files (workspace elements) while connected and disconnected. More importantly, for purposes of the ‘192 Patent analysis, the system had software to determine the preferred version of a document so as to maintain the current version on the server while allow multiple users to access the server data.

45. To achieve disconnected operation, software called “Venus” was loaded onto the client computers. The Venus software operated in “one of three states: *hoarding, emulation, and reintegration*” KS (Ex. D) at 10. In the “hoarding “ state, Venus decides which files to store locally at the client and bases this decision on many factors outlined in the KS paper at pp. 11-14. When the client is disconnected from the server, then Venus enters the “emulation” state. In the “emulation” state, Venus “performs many actions normally handled by the servers.” KS (Ex. D) at 14. Venus also maintains a record of new objects created and modifications to existing stored information:

[Venus] is also responsible for generating temporary *file identifiers* (fid) for new objects, pending the assignment of permanent fids at reintegration. . . .

Cache management during emulation is done with the same priority algorithm used during hoarding. Mutating operations directly update the cache entries of the objects involved. Cache entries of deleted objects are freed immediately, but those of other modified objects assume infinite priority so that they are not purged before reintegration.

* * *

During emulation, Venus records sufficient information to replay update activity when it reintegrates. It maintains this information in a per-volume log of mutating operations called a *replay log*. Each log entry contains a copy of the corresponding system call arguments as well as the version state of all objects referenced by the call.

KS (Ex. D) at 14.

46. During reintegration, Venus works with the server to propagate changes made during the disconnection periods and to update its memory (called “cache”):

Reintegration is a transitory state through which Venus passes in changing roles from pseudo-server to cache manager. In this state, Venus propagates changes made during emulation, and updates its cache to reflect current server state.

KS (Ex. D) at 16.

47. The algorithm for updating is as follows:

1. Upon reconnecting, Venus sends the replay log to the server.
2. The server performs the following steps:
 - a. Server locks all documents referenced in the log.
 - b. Each operation in the log is validated, then executed.
 - c. Perform data transfers.
 - d. Release all locks.

KS (Ex. D) at 16. As this algorithm makes clear, the predetermined criteria in Coda to begin synchronizing is the transmission of the replay log.

48. Claim 10 requires a “smart phone” and the use of either “an HTTP” or “an SSL” port to communicate through the firewall. With regard to the smart phone, a person of ordinary skill in the art would understand Coda as disclosing use of a smart phone. Visto does not explicitly define “a smart phone.” However, unless Visto differentiates a smart phone from a PDA or other handheld computers, Coda explicitly addresses a file system designed to handle disconnected operation with “powerful, lightweight and compact laptop computers.” KS (Ex. D) at 7. A “smart phone” is simply an extension of the same “powerful, lightweight and compact laptop computers” described and used in the Coda system.

49. Even if a smart phone is not explicitly disclosed in Coda, extending Coda to include “smart phones” would be obvious to a person of ordinary skill in the art given that the authors of Coda recognize that Coda may be extended to “environments where connectivity is intermittent . . . [including environments] that use wireless technologies such as packet radio.” KS (Ex. D) at 23.

50. Similarly, Coda discloses communicating via a firewall. The packets in Coda are encrypted and certain packets may be rejected if coded improperly. As of the time of Visto’s invention, communicating through a firewall using an HTTP or SSL port was well known. The suggestion to use an HTTP or SSL port would come from the knowledge of one of ordinary skill

in the art. That is, one of ordinary skill would know many ways to communicate through a firewall. One of ordinary skill could select any one of these methods, including use of an HTTP or SSL port. I reference a document that provides a person of skill in the art instructions on how to communicate through a firewall using such ports. For these reasons, claim 10 would be made obvious by Coda. I attach a claim chart as Exhibit B to my declaration that provides additional detail showing all elements except the HTTP/SSL requirement, and my reasons for believing that these elements do not make this invention patentable over Coda.

Lotus Notes Invalidates Claim 10

51. As set forth in greater detail in the claim chart attached as Exhibit C to this Declaration, the Lotus Notes system from the early 1990's makes Claim 10 of the '192 Patent invalid as obvious.

52. "Lotus Notes is the industry-standard client/server platform for developing and deploying groupware applications." Kevin Brown, Kenyon Brown & Kyle Brown, "Mastering Lotus Notes", SYBEX, 1995 ("Mastering Lotus Notes") (attached as Exhibit H) at Foreword. Lotus Notes is a document database. *Id.* at xxviii. Lotus Notes allows multiple people to access a database to work on it while ensuring that the database remains up to date:

Replication is a process that lets a group of people access a database on a network and work on it at the same time, while ensuring that the database remains up-to-date.

Mastering Lotus Notes (Ex. H) at 34. In terms of process, Lotus Notes compares versions and selects a preferred version:

Notes compares the different copies on the same database and saves the most recent version (according to the date the changes were made) or the version with the most changes. Subsequently, the server stores the most current version of the database for users

1 to continue to access. Replication makes all copies of a database
2 essentially identical over time. If a user makes changes in a copy
3 of the database, replication ensures that those changes are added to
4 all copies, as long as the replication options are set up to do so.

5 Mastering Lotus Notes (Ex. H) at 35.

6 53. In short, “Notes enables people in an organization to work together more
7 efficiently and productively by allowing them to share information in the same database.
8 Although people can use a database at the same time, each person can actually work with a copy
9 —or *replica*—of the database on his or her own computer.” Mastering Lotus Notes (Ex. H) at
10 35.

11 54. Furthermore, Lotus Notes “supports TCP/IP as one of it’s [sic] native protocols.”
12 Paul J. Grous, Creating and Managing a Web Site with Lotus’ InterNotes Web Publisher, The
13 View, Sept./Oct. 1995 (“Grous”) (attached as Exhibit I) at 3. By supporting TCP/IP, Notes could
14 communicate via an HTTP or an SSL port since HTTP and SSL are normally supported by an IP
15 protocol stack. Communication via an HTTP port is described as follows:

16 **Figure 6** illustrates the flow of content when published from Notes
17 into HTML. Working from left to right, you can see that content is
18 created by any Notes Client and saved in databases on any Notes
19 Server. The databases are then replicated through a firewall to
20 another Notes Server running Web Publisher. In this example, the
21 same machine is also running an HTTP server. This machine has a
22 full time TCP/IP connection to the Internet, where Web browsers
23 on all platforms can access and read the HTML pages published on
24 the HTTP server. Web Publisher automatically updates
25

1 information on your web site and keeps the HTML in synch with
2 the Notes databases.

3
4 Grous (Ex. I) at 8-9.

5 55. As shown by the above description, Lotus Notes shows:

- 6
- 7 • a system
 - 8 • a communications channel for communicating through a firewall via an HTTP or
9 SSL port;
 - 10 • software on the server for examining version information to determine whether a
11 document has been modified.
 - 12 • software on a client machine that forwards version information to indicate
13 whether a copy of the document has been modified;
 - 14 • software at the server for initiating synchronization upon receiving information
15 from a client (i.e. the predefined criteria);
 - 16 • software on the server to generate a preferred version of the document; and
 - 17 • storing the preferred version at the client and the server.

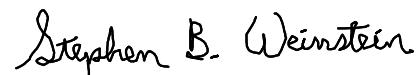
18 56. Lotus Notes is a system that discloses all elements of claim 10 except the “smart
19 phone” requirement. However, a smart phone is simply an extension of the laptop computer
20 (which may be used as a client in Lotus Notes). To the extent Visto is not distinguishing
21 between a smart phone and a PDA or other handhelds, then there is no reason Lotus Notes could
22 not be applicable to a smart phone. For these reasons, Lotus Notes renders claim 10 invalid.

23 **AMENDMENTS TO CLAIM 10 ON RE-EXAMINATION**

24 57. During re-examination, Visto made substantive amendments to Claim 10 to
25 overcome the prior art cited by the Patent Office. These amendments significantly changed the
26 scope of the claim as originally issued.

1 I declare under penalty of perjury under the laws of the United States that the foregoing is
2 true and correct.

3 Executed this 27th day of December 2005 in Summit, New Jersey.

4 
5
6 Stephen Weinstein
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25